

운영체제

-강의자료 요약-

과목명 : 운영체제
담당 교수님 : 배재환 교수님
학과 : 게임공학과
학번 : 19110171
이름 : 윤 건
제출일 : 2023.06.14.

입출력 시스템과 입출력 모듈의 개념

- 모니터나 프린터 같은 하드웨어 장치 및 입출력 모듈
- 물리적 입출력 장치가 실제로 입출력을 수행하고, 입출력 모듈은 메모리나 프로세서, 레지스터 등 내부 저장장치와 물리적 입출력 장치 사이의 이진 정보의 전송 방법 제공
- 입출력 모듈이 프로세서를 대신하여 입출력과 관련된 복잡한 일을 처리하면 입출력 채널 또는 입출력 프로세서가 되고, 단순히 프로세서의 입출력과 관련된 일을 담당하면 입출력 제어기 또는 장치 제어기가 됨

입출력 모듈의 구성

- 데이터 레지스터
- 상태*제어 레지스터
- 입출력 논리회로
- 외부장치 인터페이스 논리회로

입출력 모듈의 기능

- 내부 자원과 데이터 입출력 등 다양한 동작을 제어하고 타이밍 기능을 제공
- 입출력 모듈이 외부장치의 타이밍과 데이터 형식, 기계적인 세부 사항들을 처리하므로 프로세서는 단순히 파일 열기나 닫기만으로 장치를 제어 가능
- 프로세서가 입출력 장치에 명령을 보내려면 입출력 장치의 식별자를 주소로 지정해야 함
- 프로세서에서 명령을 전달받고, 관련된 메시지를 인식하는 기능 제공
- 버퍼링을 이용하여 전송속도 조절

입출력 방법

- 프로세서 제어 입출력
- DMA 입출력
- 입출력 채널

프로그램 제어 입출력 방법

- 프로세서 내부의 입출력 데이터와 주소 레지스터를 입출력 모듈과 연결한 형태
- 주소 레지스터와 버스 사이에서 데이터를 직접 전송할 수 있는 가장 단순한 형태
- 데이터 입력 할 때는 입출력 모듈을 거쳐 한번에 한 워드씩만 데이터 레지스터로 전송, 입출력 데이터 레지스터에서는 프로그램을 이용하여 산술 논리 연산장치로 전송
- 데이터를 출력할 때는 산술 논리 연산장치에서 입출력 데이터 레지스터로 이동, 프로그램을 이용하여 입출력 모듈로 전송

프로그램 제어 입출력 방법을 폴링 방법이라고도 함

폴링 : 상태 비트를 주기적으로 검사하여 프로세서보다 상대적으로 느린 입출력 장치의 상태를 확인

폴링방법으로 데이터를 전송할 때는 폴링 순환 수행

폴링방법에서는 순환 횟수가 데이터 전송성능에 영향. 너무 빈번하게 순환하면 입출력 동작의 진행여부를 검사하는데 시간 많이 낭비. 반대로 폴링횟수가 너무 많으면 입출력 장치가

오래 씀

입출력 채널을 이용한 입출력

입출력 채널: 프로세서와 메인 메모리를 입출력 장치에 결합하여 프로세서의 명령으로 입출력을 제어하는 장치

채널 서브 시스템은 여러 개로 구성 가능, 프로세서와 메인 메모리의 액세스 조정

채널은 사이클 스틸 사용하여 메모리에 직접 액세스하며, 각 입출력 장치는 제어장치를 사용하여 채널과 연결

제어장치는 채널과 통신하고 장치를 제어하는 기능을 수행. 유사한 장치에서는 제어장치를 하나만 사용하므로 한순간에는 장치 하나만 활성화.

입출력 채널에는 DMA 개념을 확장하여 입출력 명령어들을 실행할 수 있는 기능이 있으므로 입출력 동작을 완전히 제어하는 권한 있음

컴퓨터 시스템에서는 프로세서 가 입출력 명령어들을 실행하지 않아도 입출력 데이터를 메인 메모리에 저장

프로세서는 입출력 채널이 메인 메모리에 있는 프로그램을 실행하도록 지시하여 입출력 전송 실행. 입출력 채널은 메인 메모리에 저장된 명령어들을 실행하면서 데이터 전송 제어

채널의 종류

실렉터 채널: 어떤 장치의 입출력을 종료할 때까지 다른 장치를 실행하지 않도록 하는 것. 자기디스크 크, 자기테이프, 드럼과 같은 고속장치들은 멀티플렉서 채널을 사용하여 데이터를 고속으로 전송할 때 다른 장치들과 다중화 곤란. 이때 한번에 하나씩 처리하는 전용 채널을 이용하여 다중화 가능. 어느 한순간에 입출력 장치 중 한 장치와 하는 데이터 전송만 제어하므로 입출력 채널은 장치를 한 개 선택하여 데이터 전송

멀티플렉서 채널: 바이트단위로 시분할하여 여러 장치의 출력을 처리. 다수의 저속·중속 장치(카드 리더, 프린터)들을 연결할 때 사용. 채널-메모리 연결은 장치와 채널 사이의 데이터 전달 속도보다 더 빠른 속도로 데이터를 전달할 수 있기 때문에 속도가 느린 다수의 입출력 장치를 동시에 동작

블록 멀티플렉서 채널: 실렉터 채널과 멀티플렉서 채널의 장점을 결합한 것, 여러 대의 고속 입출력 장치들을 블록단위로 처리. 동일한 채널에서 여러 고속 장치를 활성화 가능. 채널 명령어는 장치 하나와 입출력 명령어 수행 후 별도의 지시가 없어도 자동으로 다른 장치와 명령을 수행하도록 변환되는 특징. 한 채널이 여러 입출력 장치를 시분할 방법으로 사용할 수 있지만, 어떤 때는 장치 하나에만 서비스하므로 실렉터 채널과 비슷. 그러나 멀티플렉서 채널처럼 다른 장치 서비스 완료할 때까지 기다리지 않아도 됨

커널 입출력 서브 시스템

캐싱

캐싱은 명령어와 데이터를 캐시에 일시적으로 저장하여 프로세서와 메모리 간의 액세스 속도 차이를 줄여서 컴퓨터 성능 향상 방법

캐시는 버퍼와 달리 자주 사용할 자료를 미리 복사하여 저장하는 빠른 메모리 영역. 데이터가 위치하는 유일한 장소가 버퍼라고 한다면, 다른 곳에 저장된 데이터의 복사본을 저장하는 장소는 캐시

캐싱과 버퍼링은 서로 기능이 다르지만, 때로는 한 메모리 영역을 두 가지 목적에서 사용 가능.

데이터가 캐시에 있으면 디스크 입출력을 생략하여 입출력의 효율성 높임

메모리내에버퍼를사용하여디스크입출력스케줄링을효율적으로수행하거나프로세스간에공 유해야하는파일들을위한캐시로도사용가능

스폴링

많은응용프로그램이동시에출력데이터를프린터로전송하면커널(운영체제)은각출력이다른프 로그램의출력과섞이지않고프린터로출력할수있도록관리. 각응용프로그램의출력은각각대 응되는 디스크파일에스폴링한다.스폴링시스템은응용프로그램이출력데이터작업을종료하면 그때까지 모아놓은출력데이터를프린터출력용준비큐에삽입하여스폴파일을한번에하나씩 프린터에내보낸다.이런스폴링은커널스레드로처리

오류처리

커널은입출력장치的高장이나네트워크전송오류로발생하는일시적인고장을효과적으로해결, 디스크읽기실패나망의전송오류는재실행하여문제해결.그러나중요한구성요소가영구적인 고장을 일으키면커널이문제를완전히극복하기는곤란

자료관리

커널은입출력구성에서상태정보를유지.입출력서비스를커널이아닌독립프로그램이담당하면 장치드라이버와커널의정보공유로메시지를교환하려는오버헤드가증가하는단점.장점은입출 력시스템의구조와설계가간단하고, 운영체제커널의크기가작고, 유연성이좋다는것.

입출력서브시스템은광범위한서비스를조정하며, 다음사항들을관리

장치이름과장치 액세스제어관리

장치할당과입출력 스케줄링관리

버퍼링, 캐싱, 스폴링관리

장치상태모니터링과오류처리, 고장복구 관리

디스크시스템

디스크드라이버, 프로세서, 디스크제어기로구분

디스크드라이버: 구동모터, 액세스암이동장치, 입출력헤드부분의기계적인부분담당

프로세서: 컴퓨터의논리적인상호작용, 즉데이터의위치(디스크주소)와버퍼, 판독, 기록등관리

디스크제어기

디스크드라이버의인터페이스역할

프로세서에서명령을받아 디스크드라이버동작, 디스크드라이버는 탐색seek, 기록, 판독등 명령 수행

디스크의정보는드라이버번호, 표면번호, 트랙번호등으로나누는디스크주소로참조

트랙:원형평판표면에데이터를저장할수 있는동심원을가리킨다. 자기장의간섭을줄이거나 헤드를정렬하러

고트랙 사이에일정한공간을두어 트랙구분

실린더:동일한 동심원으로구성된모든트랙, 즉동일한위치에있는 모든트랙의집합을의미. 헤드의움직임없 이액세스할수 있는드라이브의모든트랙에 해당

섹터:트랙을부채꼴모양으로나눈조각을의미. 트랙내의정보는블록을구성하고, 이블록이하드웨어적으로 크기가고정되었을때가바로섹터. 섹터는 데이터기록이나전송의기본단위로, 일반적으로

512바이트의데이터 영역으로구성.섹터는고유번호가있어디스크에 저장된데이터의위치 식별가능

파일 시스템의 개념

논리적 저장 단위인 파일을 정의하고 메모리에 매핑시키는 기능을 제공

파일을 구성하고 데이터 액세스를 관리하므로, 파일 시스템은 데이터를 실제로 저장 하는 파일과 이를 계층적으로 연결하는 디렉터리로 구성

디스크 저장소를 관리(저장 공간 할당)하는 것과 관련이 많고, 메인 메모리와 다른 매체에 저장된 파일 데이터 액세스도 포함

사용자가 파일을 생성·수정·삭제할 수 있도록 지원하며, 파일을 각 응용 프로그램에 가장 적합한 구조로 구성할 수 있도록 지원

파일 읽기, 쓰기, 실행 등을 다양하게 액세스하도록 제어된 방법도 제공

파일 시스템 설계

사용자 수, 사용자당 평균 파일 수와 크기 등 사용자 정보가 필요하고, 해당 시스템에서 실행할 응용 프로그램의 특성을 이해하여 적합한 파일 구성과 디렉터리 구조 결정해야 함

파일 시스템의 기능

파일 구성 : 사용자가 각 응용 업무에 적합하게 파일을 구성할 수 있게 함

파일 관리 : 파일의 생성, 수정, 저장, 참조, 제거, 보호 기능을 제공하며, 파일을 공유하는 다양한 액세스 제어 방법을 제공

보조 메모리 관리 : 다양한 형태의 저장장치에 입출력 지원, 2차 저장장치에 파일 저장할 공간 할당

파일 무결성 보장 : 파일에 저장된 정보를 손상하지 않도록 보장

파일 액세스 방법 제공 : 저장된 데이터(파일) 읽기, 쓰기, 실행, 이들을 여러 형태로 조합한 다양한 액세스 제어 방법을 제공

장치 독립성 유지 : 물리적 장치 이름을 사용하는 대신 기호화된 이름(예를 들어m, yDirectory:myFile.txt)으로 자신의 파일을 참조하여 장치와 독립성 유지

파일 백업과 복구 : 사고로 정보를 손실하거나 고의로 손상하는 일을 방지하려고 데이터 사본(중복)을 생성하는 백업backup과 손상된 데이터를 복구recovery할 수 있는 기능

파일 보호 : 정보를 암호화하고 해독할 수 있는 기능을 제공하여 정보의 안전과 비밀 보장

정보 전송 : 사용자가 파일 간에 정보 전송 명령을 내릴 수 있게 한다.

metadata

데이터와디렉터리외에도파일시스템크기, 새로운데이터가이미사용하는블록을덮어 쓰지않도록 저장장치의가용블록(공간), 정보(위치), 루트데이터위치와파일소유자,크기, 파일을마지막으로 수정(액세스)한시간등여러데이터정보

파일이름은보통메타데이터의일부로고려하지않음

비어있는공간비트맵, 가용블록정보와불량섹터정보포함

데이터와동일한방법으로처리

사용자가직접수정할수없으므로파일시스템의무결성유지

마운팅

시스템에서 초기에 제공하는 파일 시스템, 즉 부팅과 동시에 영구적으로 제공하는 파일 native file의 일부가 아닌 다른 정보에도 액세스해야 할 때 새로운 파일 시스템을 사용하려면 파일 시스템을 미리 제공된 파일 시스템의 디렉터리에 설치하는 절차

운영체제는 여러 파일 시스템을 마운트할 수 있는 기능 제공

여러 파일 시스템을 단일 파일 시스템에서 식별할 수 있는 파일 집합으로 결합, 운영체제에 마운트하려는 파일 시스템의 저장 위치(장치 이름)와 새로운 파일 시스템의 설치(올려놓을) 지점, 즉 마운트 포인트를 제공하면 됨

파일 시스템이 마운트되어 있으면, 사용자는 해당 파일 시스템이 다시 언마운트 될 때까지 마운트 지점에 있는 디렉터리(cdrom 등) 내용을 사용할 수 없음

파일 시스템은 마운트 테이블을 사용하여 마운트된 디렉터리들을 관리

마운트 테이블은 마운트 지점 경로명, 마운트 된 각 파일 시스템 저장하는 장치 정보 포함

파일의 개념

프로그램과 데이터 등 정보의 모음(집합)

사용자에게 프로그램, 데이터는 다른 개체이나, 파일 관리 시스템은 동일하게 파일로 처리
텍스트처럼 형태가 자유롭거나 엄격하게 제한 가능, 사용 목적에 따라 구조 특별 가능

파일 내용은 운영체제가 물리적 장치에 저장.

사용자 관점에서 파일은 논리적으로 저장되는 기본 단위로, 프로그램이나 데이터 될 수 있음. 이 논리적 파일을 실제 저장장치에 매핑시키는 작업은 운영체제가 담당

파일의 세분화

파일의이름

디렉터리의루트에서위치까지경로포함

특정파일의경로명은디렉터리이름과파일이름으로구성.

모든경로명을표시하여파일을참조하는절대경로가너무길어불편할때는작업디렉 터리에서상대적 위치를지정하는상대경로로참조가능

파일의속성

시스템이파일을관리하는데필요한정보

대부분속성을포함하는파일제어블록은디스크에저장,파일정보는메인메모리에유 지하여파일을 열때탐색시간줄임

파일속성은파일시스템에따라구성다름

파일속성의일반적구성

파일의유형

파일의유형으로파일의내부구조형태짐작가능

운영체제는파일시스템이지원가능한파일구운영체제는파일시스템이지원가능한파일구조정의, 해당파일을다룰수있는특별한연산기능제공

파일의세가지유형

일반(정규)파일

가장일반적인파일과데이터를포함하는데사용,텍스트나이진형태

디렉터리파일

모든유형의파일에 액세스할수있는정보 포함하나,실제파일데이터는 포함하지않음

특수파일

시스템장치를정의하거나프로세스로생성한 임시파일로파이프라고 하는FIFO(선입선출),블록,
문자이에해당

운영체제가여러파일의구조를지원하면크기가커져복잡해진다는단점

너무적은파일구조를지원하면프로그래밍어려움

파일이름에마침표를넣어서구분하여파일유형표현

파일이름은크게순수이름과확장자로구성

순차엑세스

파일에있는정보는레코드단위의순서로처리하는것이가장일반. 파일에서대부분의 동작은읽기와
쓰기.순차엑세스에서읽기동작은파일의다음부분을읽은후자동으로 파일포인터증가시킴

쓰기동작은파일의끝에내용추가하고,포인터를쓴내용(파일의새로운끝)의끝으로이 동.순차파일
은일반적으로데이터입력파일에는사용하지않고프로그램의임시작업 파일로사용.이런파일은시
작위치로재설정가능,어떤시스템에서프로그램은정수n 개의레코드를앞뒤로건너뛰기가능

파일의테이프모델기반

인덱스 순차 액세스

직접 액세스를 기반으로 디스크의 물리적 특성에 따라 인덱스를 구성하여 탐색, 포인터 사
용하여 파일 액세스

큰 파일도 적은 입출력으로 탐색 가능, 인덱스도 1~2차 인덱스 파일로 구성 처리 가능

파일에서 특정 항목을 찾으려면, 먼저 인덱스 탐색, 그런 다음 포인터 사용하여 파일 직접
엑세스해서 원하는 항목 찾아냄

디렉터리의 개념

디렉터리를 유지하고 관리하여 디스크 등에 저장된 파일을 관리

디렉터리의 구분

장치 디렉터리 : 각 실제 장치에 저장. 장치에 있는 파일의 물리적 속성, 파일의 위치, 파
일의 크기와 할당 과정 등을 나타냄

파일 디렉터리 : 모든 파일의 논리적 구성으로 이름, 파일 유형, 소유한 사용자, 계정 정
보, 보호 액세스 코드 등을 기술

파일 시스템에서 다른 파일들의 이름과 위치 정보(파일 인덱스)를 담은 파일로, 다른 파일
들과 달리 사용자 데이터 저장하지 않음

장치의 범위를 확장할 수 있고 다른 디스크 장치들을 포함 가능

디렉터리에 있는 정보 중 일부는 사용자나 응용 프로그램이 이용 가능하나, 대부분 시스템
루틴이 사용자에게 간접적으로 제공

선형 리스트를 이용한 디렉터리 구현

디렉터리에 파일 이름, 포인터들의 선형적 리스트 구성하여 파일의 생성과 삭제 등을 실행
선형 탐색을 해야 파일을 찾을 수 있어 오버헤드가 증가하므로 시스템 성능 떨어뜨릴 수 있는 단점

사용된 디렉터리 정보를 저장하는 소프트웨어 캐시를 구현하여 정보를 매번 디스크에서 읽어 오지 않도록 하여 문제 해결(캐시적중).

리스트 정렬하여 이진 탐색 방법 사용

평균 탐색 시간은 줄일 수 있으나, 리스트가 정렬 상태를 유지하려고 하면 파일 생성과 삭제 복잡

이진 연결 트리 사용하여 문제 해결

해시 테이블hash table을 이용한 디렉터리 구현

해시 테이블을 이용하여 파일 이름을 제시하면, 해시에서 값을 얻어 리스트를 직접 액세스하도록 디렉터리 구현 가능

디렉터리 탐색 시간을 줄이면서 성능을 개선하기에 많이 사용

둘 이상의 파일 이름이 같은 위치를 지정할 때는 충돌 발생 가능성, 보완하면 쉽게 삽입 · 삭제 가능

해시 테이블의 크기가 고정되고 크기에 따라 해시 기능이 제한을 받는다는 문제점

해시 테이블의 크기는 파일의 크기 정보 활용

해시 테이블에서 충돌을 해결하는 방법으로 체인 오버플로 해시 테이블 사용

각 해시 항목은 하나의 값이 아니라 연결 리스트가 되고, 새로운 항목을 연결 리스트에 추가하여 충돌 해결

이름을 찾으려면 연결 리스트를 살펴봐야 하기 때문에 찾기 작업은 다소 늦어지나, 전체 디렉터리를 선형으로 찾는 것보다는 훨씬 빠름

디스크 할당 방법의 비교

연속 할당

파일의 직접 액세스를 지원하며, 연결 할당으로 순차 액세스 지원

작은 파일에 효율적, 평균 성능이 아주 좋음

연결 할당

메모리에 다음 블록 주소를 보관할 수 있으며, 이를 직접 읽을 수 있음

순차 액세스는 매우 쉬우나, 직접 액세스는 i번째 블록을 읽으려고 디스크를 i번 읽어야 할 때도 있음

인덱스 할당

인덱스 블록이 메모리에 있으면 직접 액세스 가능.

메모리에 인덱스 블록을 보관하려면 메모리가 많이 필요

2단계 인덱스는 인덱스 블록을 두 번 읽어야 함

인덱스 할당의 성능은 인덱스 구조와 파일의 크기, 원하는 블록의 위치가 좌우.

파일이 작으면(블록 3~4개 정도) 연속 할당을 사용하고, 파일이 크면 자동 인덱스 할당으로 변환하는

방법을 사용하기도 함

인덱스 블록(그룹핑)

빈 블록의 포인터를 인덱스 블록에 보관하며, 이들은 서로 연결

빈 블록 주소를 포함하여 인덱스 역할을 하므로, 첫 번째 빈 블록 내에 빈 블록 n개의 번지 저장

이것의 처음 n-1개는 실제로 비어 있고, 마지막 한 개는 다른 빈 블록 n개의 주소 포함하는 다른 블록의 디스크 주소

사용 가능한 블록 주소를 여러 개 쉽게 찾을 수 있게 함

파일 보호의 필요성

컴퓨터 시스템은 다수의 사용자가 사용하므로 정보를 저장한 파일은 물리적인 손상이나 부적합한 액세스에서 보호해야 함. 특히 파일을 물리적인 손상에서 보호(신뢰성 보호)해야 함.

보통은 디스크에서 테이프로 파일을 자동 복사하여 파일이 불의의 사고로 손실되거나 손상되는 것을 방지

파일 시스템은 하드웨어의 문제점인 전원 장애, 헤드 파손, 먼지, 온도, 고의적인 파괴 행위로 손상을 받거나 우연한 사고로 제거될 수 있음

파일 시스템 소프트웨어의 오류로 파일 내용을 분실할 수도 있다. 단일 사용자 시스템이라면 단순히 사용 디스켓을 잘 관리하거나 잠금 장치를 이용하면 되지만 다수 사용자 시스템에서는 다른 방법이 필요

파일 명명file naming

액세스하려는 파일을 명명할 수 없는 사용자를 액세스 대상에서 제외하여 파일 보호 가능.

파일 이름은 보통 기억하기 쉬운 글자를 많이 사용하므로 추측도 가능

암호password(비밀번호)

각 파일에 암호를 설정하여 보호

암호를 임의로 정하고 자주 변경하면 파일의 액세스를 효율적으로 제한 가능

파일별로 독립된 암호가 있어 기억해야 할 암호가 너무 많아 비현실적

모든 파일의 암호가 하나이면 한 번만 노출되어도 모든 파일에 액세스할 수 있다는 문제
세밀한 수준의 보호를 제공하려면 다중 암호를 사용

액세스 제어access control

사용자에 따라 액세스할 수 있는 파일이나 디렉터리 리스트를 두어 사용자 신원에 따라 서로 다른 액세스 권한 부여

액세스 요구가 발생하면 운영체제가 이 액세스 리스트를 참조하여 액세스 여부 결정 방법으로 보호

네트워크와분산시스템의개념

운영체제에서분산처리는컴퓨터사용자간에서로데이터를교환하여처리할수있도록 네트워크로 상호연결한것

네트워크로연결한시스템은사용자의엑세스를제어하여편리하게자원공유

네트워크로연결한시스템은분산시스템과다중처리시스템으로구분

분산시스템:메모리와클록을공유하지않고지역메모리를유지하는프로세서로구성 서로독자적으로동작

다중처리시스템:하나이상의프로세서로구성되며,프로세스들이메모리와출력을공유

병렬처리시스템이라고도함

네트워크의 개념

서로 독립된 시스템 몇 개가 적절한 영역 안에서 속도가 빠른 통신 채널을 이용하여 상호 통신할 수 있도록 지원하는 데이터 통신 시스템

컴퓨터 네트워크는 1960년대 사이트 간 효율적 통신 위해 학교 연구 프로젝트로 탄생 광범위한 사용자 모임 간에 하드웨어나 소프트웨어를 편리, 경제적 공유 지원

알파넷ARPANET :최초로 개발된 네트워크(1968년에 처음 작동)

사용자가 원거리의 하드웨어나 소프트웨어 자원에 액세스할 수 있는 기능 제공

네트워크 시스템 구성하는 방법 : 강결합 약결합

원격 프로시저 호출의 개념

분산 시스템과 단일 시스템의 가장 큰 차이는 프로세스 간 통신이다. 단일 시스템에서 프로세스들은 공유 메모리로 입출력하여 서로 통신 가능하나, 분산 시스템은 공유 메모리 없음

분산 시스템에서 프로세스 간 통신은 클라이언트와 서버의 요청·응답 형태로 구현하는 대표적인 예

Birrell과 Nelson이 제안

한 컴퓨터에서 실행하는 프로세스를 다른 컴퓨터에서 실행하는 프로세스의 프로시저(또는 함수)

가 호출할 수 있게 하는 것으로, 클라이언트/서버 모델을 전제로 함

클라이언트와 서버 사이에 프로시저를 호출하는 프로그램을 허용하여 지역 프로시저를 호출해서 지역적으로 분산된 원격 프로시저 호출

클라이언트에 있는 프로세스가 서버에 있는 프로시저를 호출하면 클라이언트에 있는 프로세스는 중단하고 서버에 있는 호출된 프로시저 실행

함수의 반환 값이 네트워크 이용하여 클라이언트로 전송

분산 시스템의 구조

저렴한 노드 여러 개를 운영체제 하나가 제어할 수 있도록 구현, 강력한 시스템 구성

네트워크로 연결된 여러 노드에 프로그램 하나를 분산하여 실행하면서 마치하나의 프로그램처럼 동작

여러 사용자가 자원 공유하여 대규모 작업 지원하므로 다양한 사용자에게 서비스 가능

각 프로세서는 초소형 프로세서부터 워크스테이션, 소형컴퓨터, 대형컴퓨터까지 다양

네트워크운영체제의주요기능

자원공유: 네트워크에 연결된 서버와 클라이언트 간 하드디스크나 프린터 등 자원

액세스 권한 부여: 사용자는 원격 사이트의 자원 사용할 수 있도록 액세스 가능해야 함

자격 있는 사용자만 특정 자원을 사용하도록 제한 가능

호스트 컴퓨터의 하드웨어 사항을 몰라도 되지만, 원격 시스템의 명령은 알아야 함

파일 전송: 한 컴퓨터에서 다른 컴퓨터로 데이터 전송

데이터 보호: 사용자별 적합한 권한(읽기 권한, 쓰기 권한 등) 설정하고 데이터 관리 · 보호 하여 서버에 접근하는 클라이언트 사용자가 서버를 사용할 권한이 있는지 인증

관리 제어: 각 클라이언트의 네트워크 이용 정보와 네트워크에서 발생할 수 있는 여러가지 문제 해결하고 조절하는 관리 기능 제공

클라이언트/서버 방법

네트워크와 독립적이며, 네트워크에 접속하지 않은 컴퓨터 시스템은 자신의 환경에 맞게 변경 · 관리 가능

네트워크에 연결된 컴퓨터가 각각 임무, 즉 클라이언트는 클라이언트, 서버는 서버로서 역할 수행 방법

클라이언트/서버 모델이 워크스테이션 모델이면 클라이언트는 예외적으로 동시에 사용 가능한 명이 사용 가능

중앙에서 관리하는 형태로 응용 프로그램에 전용 파일 서버를 허용

파일 서버는 보안과 자원 액세스를 제공하는 시스템의 핵심

개별 워크스테이션(클라이언트)은 파일 서버에 있는 사용 가능한 자원 액세스

분산 운영체제의 제어하에서의 작업 수행

데이터 이동 data migration : 사이트 A에 있는 사용자가 사이트 B에 있는 파일 등 데이터에 액세스하려면 데이터를 다음 두 가지 방법으로 전송해야 함, .파일의 일부분만 액세스하는 응용 분야에서는 두 번째 방법이 더 유리. 그러나 파일 대부분을 액세스할 때는 파일 전체를 복사하는 것이 더 효율적

파일 전체를 사이트 A로 전송하는 방법. 이때 해당 파일에서는 지역적으로 모두 액세스. 사용자가 더

는 파일을 액세스할 필요가 없을 때 수정했는지 모르나, 파일 복사본은 사이트 B로 되돌려 보냄

실제로 작업에 필요한 부분만 사이트 A로 전송하는 방법. 사용자가 더는 그 파일에 액세스할 필요가 없으면 수정된 파일 일부분만 사이트 B로 되돌려 보냄(요구 페이징 방법과 비슷)